Article

Integration of Multiple Bayesian Optimized Machine Learning Techniques and Conventional Well Logs for Accurate Prediction of Porosity in Carbonate Reservoirs

Saad Alatefi [1,*] **, Reda Abdel Azim** [2] **, Ahmad Alkouh** [1] **and Ghareb Hamada** [2]

[1] Department of Petroleum Engineering Technology, College of Technological Studies, PAAET, P.O. Box 42325, Kuwait City 70654, Kuwait; ab.alkouh@paaet.edu.kw

[2] Petroleum Engineering Department, American University of Kurdistan, Erbil 42003, Iraq; reda.abdulrasoul@auk.edu.krd (R.A.A.); ghareb.hamada@auk.edu.krd (G.H.)

[*] Correspondence: se.alatefi@paaet.edu.kw

Abstract: The accurate estimation of reservoir porosity plays a vital role in estimating the amount of hydrocarbon reserves and evaluating the economic potential of a reservoir. It also aids decision making during the exploration and development phases of oil and gas fields. This study evaluates the integration of artificial intelligence techniques, conventional well logs, and core analysis for the accurate prediction of porosity in carbonate reservoirs. In general, carbonate reservoirs are characterized by their complex pore systems, with the wide spatial variation and highly nonlinear nature of their petrophysical properties. Therefore, they require detailed well-log interpretations to accurately estimate their properties, making them good candidates for the application of machine learning techniques. Accordingly, a large database of (2100) well-log records and core-porosity measurements were integrated with four state-of-the-art machine learning techniques (multilayer perceptron artificial neural network, MLP-ANN; Gaussian process regression, GPR; least squares gradient boosting ensemble, LS-Boost; and radial basis function neural network, RBF-NN) for the prediction of reservoir porosity. The well-log data used in this study include sonic acoustic travel time, Gamma-ray, and bulk density log records, which were carefully collected from five wells in a carbonate reservoir. This study revealed that all the artificial intelligence models achieved high accuracy, with R-squared values exceeding 90% during both the training and blind-testing phases. Among the AI models examined, the GPR model outperformed the others in terms of the R-squared values, root-mean-square error (RMSE), and coefficient of variation of the root-mean-square error (CVRMSE). Furthermore, this study introduces an artificially intelligent AI-based correlation for the estimation of reservoir porosity from well-log data; this correlation was developed using an in-house, Fortran-coded MLP-ANN model presented herein. This AI-based correlation gave a promising level of accuracy, with R-squared values of 92% and 90% for the training and blind-testing datasets, respectively. This correlation can serve as an accurate and easy-to-use tool for porosity prediction without any prior experience in utilizing or implementing machine learning models.

Keywords: porosity AI-based correlation; carbonate reservoirs; machine learning techniques; Gaussian process regression; least squares gradient boosting ensemble

1. Introduction

Porosity, which is a measure of hydrocarbon reservoir storage capacity, is a key petrophysical property that usually aids in the decision-making process during the exploration, development, and production phases of oil and gas fields. Therefore, accurate estimation of this property is of paramount importance to the petroleum industry.

The ideal method for measuring porosity is through the laboratory testing of core samples taken from subsurface hydrocarbon zones. This option may not always be feasible due to its high cost, which makes it challenging to obtain measurements for every well or every

section of a well within a given field. Therefore, there is a need for methods to estimate the porosity from readily available conventional well logs, which is a more economical option since these logs are routinely collected during drilling and well-completion operations. Thus, various correlation and nonparametric regression techniques have been developed to estimate reservoir porosity from well logs [1–9]; however, these correlations may have limitations (e.g., they require prior knowledge of the lithology and pore fluid types) and uncertainties in accuracy, especially in complex reservoirs with significant heterogeneity.

Accordingly, it is crucial to continually improve and develop new porosity estimation techniques to ensure accurate reservoir characterization and improve the reservoir management process. Therefore, in recent years, researchers have shifted their focus towards utilizing machine learning techniques to enhance porosity estimation accuracy because these techniques are well-equipped to handle the high nonlinearity in input data—these are data-driven techniques and therefore do not require prior knowledge of lithology or pore fluid types [10–12]. By utilizing large volumes of well-log data and core samples, these techniques can provide more accurate porosity estimates, even in complex reservoirs with significant heterogeneity [13–15]. Furthermore, machine learning models can also incorporate data from multiple sources, such as seismic features and well-log data, to provide a more comprehensive understanding of the reservoir properties and improve the porosity estimation accuracy [16,17]. In the literature, artificial neural networks [18–24], support vector machines [25–27], fuzzy logic [28–30], and neuro-fuzzy [31–33] are all examples of machine learning techniques that are often used for reservoir characterization, including porosity estimation. However, the majority of the previous studies are limited in the number and range of data used, making them prone to deficiencies, such as an inadequate prediction of porosity in heterogeneous reservoirs similar to the carbonate reservoir discussed in this study.

Consequently, this study presents the use of a large database of carbonate-reservoir core-porosity measurements and well-log records, including sonic acoustic-travel-time, Gamma-ray, and bulk-density logs, in the utilization of four state-of-the-art machine learning techniques (multilayer perceptron artificial neural network, MLP-ANN; Gaussian process regression, GPR; least squares gradient-boosting ensemble, LS-Boost; and radial basis function neural network, RBF-NN) for the prediction of reservoir porosity. The database used in this study comprises (2100) conventional well-log records and core-porosity measurements that were carefully gathered from five wells in a carbonate reservoir located in the Middle East region. Carbonate reservoirs, in particular, are characterized by their complex pore systems and heterogeneity (i.e., the wide spatial variation in their petrophysical properties). Therefore, they require detailed well-log interpretations to accurately estimate their properties, making them good candidates for the application of machine learning techniques. Furthermore, since the empirical correlation used for well-log interpretation may have poor accuracy in such heterogeneous reservoirs, we propose herein a smart AI-based correlation for the well-log interpretation of a carbonate reservoir. The proposed correlation is built using an in-house, Fortran-coded multilayer perceptron artificial neural network model developed by Azim [34]. This smart correlation could serve as an accurate and easy-to-use tool for porosity prediction without any prior experience in utilizing or implementing machine learning models. Furthermore, it can be used for estimating porosity profiles for uncored wells in the same reservoir, saving time and the cost of extensive coring measurements in newly drilled wells.

2. Data Acquisition and Analysis

2.1. Porosity Database

The main objective of this study was to establish a reliable porosity model for a carbonate reservoir in the Middle East region, using a large database of core-porosity measurements and conventional wireline log records. The reservoir under study is an Upper Jurassic carbonate reservoir located in the Arab-D formation of the Ghawar field. The studied reservoir is mainly limestone with interbeds of dolostone. In general, the Arab-D

formation of the Ghawar field is grouped into several litho-facies types, including seven limestone and four dolomite rock types. These rocks have been exposed to different diagenetic processes, including dolomitization, compaction, leaching, recrystallization, and fracturing, resulting in high spatial variation (both horizontally and vertically) in the reservoir's petrophysical properties.

A total of 2100 well-log records were collected from five wells (Wells A, B, C, D, and X) and each dataset contained the following independent parameters:

1. Sonic acoustic travel time log (DT), μs/ft;
2. Bulk density log (RHOB), g/cm^3;
3. Gamma-ray log (GR), API unit.

The presented well-log records were calibrated to core data through depth-shifting in order to make sure that the porosity estimated from the well-log data was cross-checked with experimentally measured porosity.

The range of statistical parameters for the well-log and core-porosity data under study is presented in Table 1, which demonstrates the range of variations for dependent and independent parameters for all (2100) datasets used in this study.

**Table 1.** Statistical parameters of the porosity database used.

Statistical Parameter	Sonic Log	Bulk Density Log	Gamma Ray Log	Porosity
	μs/ft	g/cm^3	API	Fraction
Maximum	70.81	3.07	82	0.2154
Minimum	42.4	2.6	7.8	0.0003
Mean	49	2.8	29.8	0.0377
Standard Deviation	4.61	0.096	11.4	0.039
Coefficient of Variation	0.092	0.034	0.38	1.04

The porosity database presented in Table 1 was utilized to train and cross-validate the developed machine learning models (GPR, LS-Boost, RBF-NN, and MLP-ANN), using a 5-fold cross-validation technique (unless otherwise stated) to minimize any overfitting issues.

For each well, 80% of the data were randomly used for training/cross-validation (65% for training and 15% for validation), while the remaining 20% of the data were kept as blind-testing (unseen) data in order to examine the model's generalization ability. Accordingly, a total of 1542 well-log records were used for training/cross-validation and a total of 386 well-log records were used for blind testing. Figure 1A presents the well-log records used for training, while the testing data are shown in part (Figure 1B) of the same figure. The data were normalized in order to avoid any scale issues. Thus, the data were mapped to the maximum and minimum values of the input variables using the following equation:

$$x_n = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

where x is the input variable, x_{min} is the minimum of the input variable, and x_{max} is the maximum of the input variable.
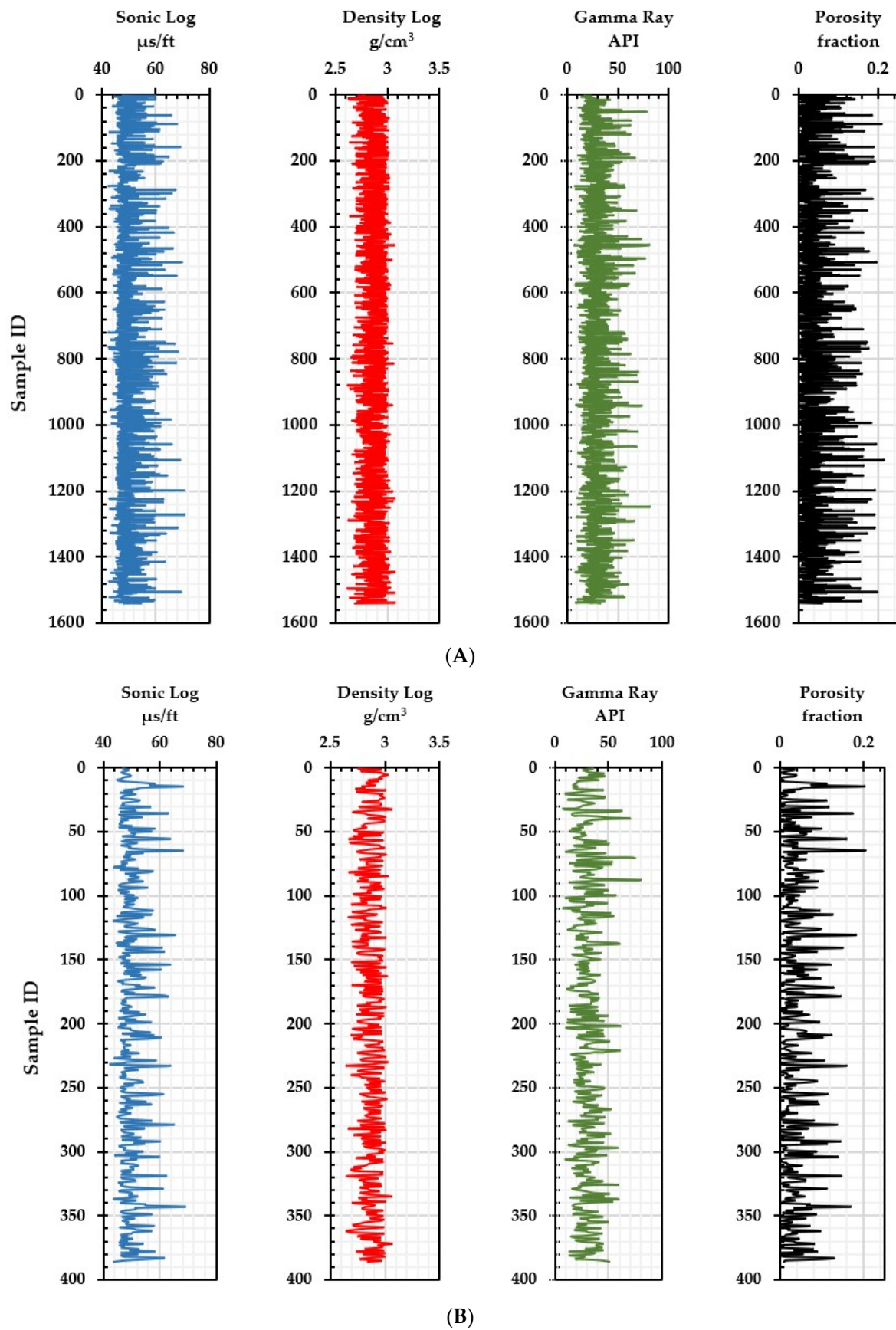
Figure 1. Well-log records of (DT, RHOB, and GR) used in this study: (**A**) used as training data; (**B**) used as (unseen) blind-testing data.

2.2. Performance Indicators

The prediction accuracy of the studied models was assessed using various statistical indicators, such as the root-mean-square error (RMSE), the coefficient of variation of the root-mean-square error (CVRMSE), and the coefficient of determination R^2. These indicators are presented by Equations (2)–(4) as follows:

$$\text{RMSE} = \frac{\sqrt{\sum_{i=1}^{n}(y_i - \check{y}_i)^2}}{n} \tag{2}$$

$$\text{CVRMSE} = \frac{1}{\overline{y}} \frac{\sqrt{\sum_{i=1}^{n}(y_i - \check{y}_i)^2}}{n} \tag{3}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \check{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \tag{4}$$

where $\check{y}_i$ is the estimated response and $\overline{y}$ is the average true response.

3. Methodology

Using four artificially intelligent models, including the multilayer perceptron artificial neural network (MLP ANN), Gaussian process regression (GPR), least squares gradient boosting ensemble (LS-Boost), and radial basis function neural network (RBF-NN), this study aimed to create a general intelligent model that can accurately predict the reservoir porosity profile in carbonate reservoirs. The MATLAB software (R2021a version, MathWorks Inc., Natick, MA, USA) was used for the coding and implementation of all the models except MLP-ANN, which is an in-house, Fortran-coded model developed by Azim [34]. Herein, we provide a brief overview of each intelligent model developed in this research, as well as the Bayesian optimization algorithm that was used to fine-tune the model's hyperparameters.

3.1. Gaussian Process Regression

Gaussian process regression is a powerful machine learning technique based on the Bayesian theory [35]. This approach is particularly suitable for addressing small sample sizes, complex nonlinear problems, and high-dimensional data [36,37]. Unlike linear regression methods that rely on deterministic variables alone, GPR leverages a set of random variables with joint Gaussian distributions characterized by mean and covariance functions.

Assume an input and target vector $\{x_i, y_i\}$ with m training data points where $i = 1, 2, \ldots, m$. Then, the model can be defined as follows:

$$y_i = f(x_i) + \epsilon_i \tag{5}$$

where ϵ_i is a Gaussian noise with zero mean and variance σ_n^2 and $f(x_i)$ is the learning function. The output vector y_i can be presented with a Gaussian distribution as follows [37]:

$$y \sim \mathcal{N}\left(0, K(X, X) + \sigma_n^2 I\right) \tag{6}$$

where $K(X, X)$ is the kernel function (or covariance function). Various types of kernel functions can be utilized in the Gaussian process, such as squared exponential, exponential, Matern 3/2, and Matern 5/2 [35]. For instance, the squared exponential function is as follows:

$$k(x_p, x_q) = \sigma_f^2 \exp\left(-0.5 \sum_{i=1}^{m} (x_{pi} - x_{qi})^2 / \sigma_l^2\right) \tag{7}$$

where σ_f^2 and σ_l^2 are the signal variance and the length scale of the kernel function. For any testing data point x^*, the joint Gaussian distribution of the true response values and the estimated ones is given as [37]:

$$\begin{bmatrix} y \\ f(x^*) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} k(X,X) + \sigma_n^2 I & k(X,x^*) \\ k(x^*,X) & k(x^*x^*) \end{bmatrix}\right) \tag{8}$$

This will give a predicted mean $f(x^*)$ and a variance $V(x^*)$ of the learning function [37]:

$$f(x^*) = k^{*T}\left(K + \sigma_n^2 I\right)^{-1} y = k^{*T}\alpha \tag{9}$$

$$V(x^*) = k(x^*,x^*) - k^{*T}\left(K + \sigma_n^2 I\right)^{-1} k^* \tag{10}$$

The parameters $= \left[\sigma_n^2, \sigma_f^2, \sigma_l^2\right]$ are the Gaussian process hyperparameters optimized using the Bayesian optimization algorithm in this study.

3.2. Least Square Gradient Boosting Ensemble

Ensemble learning combines multiple models to enhance prediction accuracy and robustness in supervised machine learning. This approach is particularly useful when dealing with complex datasets or when individual models have limitations [38]. In this technique, a meta-learner integrates several regression machine learning methods and weighs each based on their performance to create a more reliable ensemble model that enhances prediction accuracy and reduces overfitting. Popular ensemble techniques include bagging, boosting, and random forests [39]. In addition, different types of individual learners can be ensembled, including decision trees, neural networks, support vector machines, and logistic regression models [40,41]. The variety of methods will lead to varying regression performances, enhancing the overall performance of the ensemble technique. However, it is important to note that ensemble learning may require more computational resources and time for training compared to individual models [38,42].

The LS-Boost ensemble used in this study employs decision trees as individual learners to reduce global error. Individual learners are trained sequentially on the testing dataset and fitted with residuals of errors. With each iteration, a new learner is fit to improve prediction accuracy by minimizing the differences between response values and aggregated predicted values. The least square boosting ensemble method is presented in Algorithm 1 as reported by Friedman in [43].

Algorithm 1: LS-Boost Algorithm

Define x_i and y_i as explainable variables and M as the number of iterations.

Define the training set $\left\{\left(x_i, y_j\right)\right\}_{i=1}^n$, a loss function as $L(y,F) = \frac{(y-F)^2}{2}$ and $F_m(x)$ as the regression function.
Initialization: $F_0(x) = \bar{y}$
For $m = 1$ to M do:
    $\tilde{y}_i = y_i - F_{m-1}(x_i)$ for $i = 1, 2, \ldots, N$
    $(\rho_m, \alpha_m) = \text{argmin}_{\rho,\alpha} \sum_{i=1}^{N}\left[\tilde{y}_i - \rho h(x_i; \alpha)\right]^2$
 $F_m(x) = F_{m-1}(x) + \rho_m h(x; \alpha_m)$
End

3.3. Multi-Layer Perceptron Artificial Neural Network

An artificial neural network (ANN) is a powerful computational technique that emulates the complex structure of the human brain system. The ANN design aims to replicate how our brains acquire knowledge, comprehend information, and provide solutions to challenging problems through learnability techniques. ANNs are composed of multiple interconnected processing nodes (neurons) that work together to process and analyze complex datasets. That is, an ANN is composed of an input layer, a single layer (or multi-

ple hidden layers), and an output layer, and these layers are connected through synaptic weights. Thus, ANNs are based on the concept of a weighted sum of inputs, where each input is multiplied by a weight and then summed together. This weighted sum is then passed through an activation function, which determines the output of that particular node. The process is repeated for each node in the network, with the output of one node becoming the input for another node until the final output is produced. Some popular activation functions include sigmoid, ReLU (rectified linear unit), and tanh (hyperbolic tangent) [44]. Once the network outputs are calculated, they are compared to the true target values to calculate the error, and this error is used to adjust the weights of the network through a process called backpropagation. Through this process, ANNs can identify hidden trends within the data that may be difficult for conventional regression techniques to detect. Backpropagation is a common process used to train ANNs by adjusting the weights and biases of the network to minimize the divergence between the predicted and the actual outputs using gradient-based algorithms. This process is repeated over multiple iterations until the error is minimized to an acceptable level, at which point the model is considered trained and ready for testing and evaluation.

The success of an ANN model largely depends on the quality and quantity of data used for training [45,46], as well as the appropriate selection of network architecture (i.e., the number of hidden neurons and hidden layers) and the tuning of hyperparameters (i.e., determination of the optimum sets of network weights and biases). A typical MLP network is shown in Figure 2.
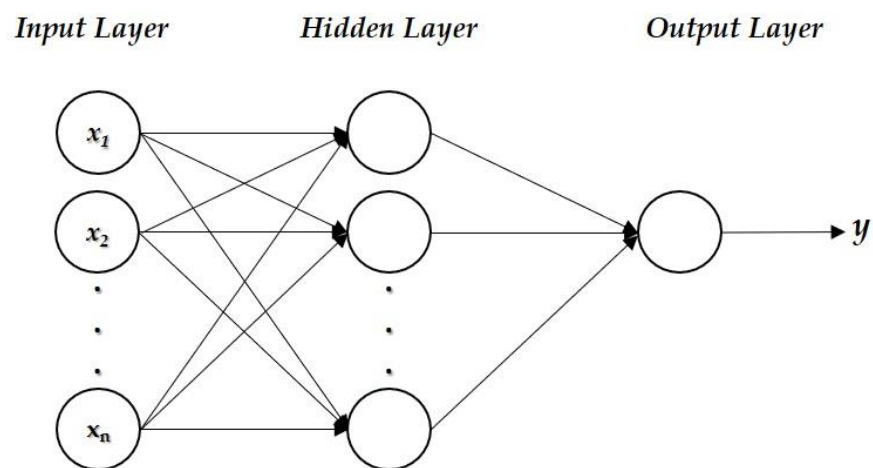
Figure 2. Typical MLP neural network.

In this study, we utilized an in-house, Fortran-coded artificial neural network model developed by Azim [34]. The in-house code uses the backpropagation algorithm for supervised learning. The in-house code structure can be summarized as follows:

1. Set the initial weights and biases of the network in a random manner.
2. Start the forward pass and calculate the network output utilizing the input vector, weights, biases, and transfer functions.
3. Compare the network output to the true response and calculate the global error using the following formula:

$$Error = \frac{\sum_{1}^{n_1}\sum_{1}^{n_2}\left(y_t - y_p\right)}{n_1 \cdot n_2} \tag{11}$$

where n_1 is the total number of training instances and n_2 is the number of neurons in the output layer. The y_t and y_p are the true and predicted output values, respectively [34].

4. Propagate back to adjust the weights and biases using one of the following gradient-based algorithms, scaled conjugate gradient, one-step secant, or Levenberg–Marquardt

algorithm, in order to decide the amount and direction of the weight change. In addition, the following convergence technique is used to speed up the network through adding an acceleration factor as follows [47]:

$$w(t + 1) = w(t) + \beta[\Delta w(t)] + \alpha[w(t - 1)]$$ (12)

where α is the momentum constant, w is the weight value, Δw is the weight change, t is the training epoch, and β is the learning constant. The constants α and β are employed to increase the step size and decrease abrupt gradient changes, and these learning and momentum constants are confined between 0 and 1 [34].

5. Use the new set of weights and biases to recalculate the network output by repeating steps 2 to 4.
6. Report the final optimized set of weights and biases once the model reaches a predefined accuracy level or a maximum number of iterations.

In recent studies, Reda Abdel Azim [34,48,49] concluded that a well-trained MLP neural network can be conveniently converted into an AI-based mathematical model using the optimized weights and biases of the network, and thus he developed a set of robust AI-based correlations for different petroleum applications including the fields of fluid properties, drilling operations, and production optimization. In this study, we followed the same approach to propose an AI-based explicit mathematical formula for the accurate prediction of reservoir porosity (as shown in Section 4.2).

3.4. Radial Basis Function Neural Network

The radial basis function neural network (RBF-NN) is another kind of artificial neural network that was utilized in this study, which uses radial basis functions as activation functions. These functions measure the distance between a given input and a center point, with the output being the highest when the input is closest to the center point. This type of ANN is commonly used in function approximation and interpolation, clustering, and classification tasks [46]. The RBF neural network is based on the concept of a hidden layer of nodes, each representing a radial basis function. The output of each node in the hidden layer is then combined and passed through a linear combination to produce the final output. The main difference between RBF and multi-layer perceptron ANN is that RBF consists of only one hidden layer; additionally, in RBF, the weights are calculated using methods other than backpropagation, such as orthogonal least-squares approximation [46,50].

The output of any RBF function in the hidden layer can be estimated based on the Euclidean distance between the center of the Gaussian function and the input vector x, as shown in Equation (13):

$$\varnothing_i \left(\left\| x - c_j \right\| \right) = \exp \left[\frac{\left\| x - c_j \right\|^2}{2\sigma_j^2} \right]$$ (13)

where σ_j and c_j are the spread and center of the Gaussian RBF function, respectively.

The final output of the RBF network is a weighted sum of all hidden-layer RBF nodes, as shown below:

$$Y = \sum_{j=1}^{n} w_{ij} \times \varnothing_i \left(\left\| x - c_i \right\| \right)$$ (14)

where w_{ij} is the weight connecting the jth neuron in the hidden layer to the ith neuron in the output layer.

Standard RBFN training involves two stages: (1) determining the centers and spreads using clustering techniques, and (2) finding the connection weights between the hidden layer and output layer by minimizing the mean squared error over the entire dataset.

3.5. Bayesian Optimization Algorithm

This study utilized the Bayesian optimization method to tune the hyperparameters of machine learning models (i.e., Gaussian process regression, least square boosting ensemble,

and RBF-Neural Network) for improved cross-validation scores. Bayesian optimization is particularly useful for computationally expensive function evaluations by reducing the time taken to achieve global minima within solution spaces [38]. The exploration and sampling of search space rely on prior beliefs about the problem according to Bayes' theorem, which states that "posterior probability of a model *M* given evidence *E* is proportional to likelihood of *E* given *M* multiplied by prior probability of *M*", as shown in the following formula:

$$P(M|E) \; \alpha \; P(E|M)P(M) \tag{15}$$

A surrogate model, e.g., the Gaussian process, approximates the objective function and selects samples from the solution space using acquisition functions, such as expected improvement and maximum probability of improvement [51]. Algorithm 2 outlines the Bayesian optimization approach [38].

Algorithm 2: Bayesian optimization

For $t = 1, 2, \ldots$ do
Find x_t by optimizing the acquisition function over the Gaussian Process (GP)
$$x_t = \text{argmax}_x u(x|\mathcal{D}_{1:t-1})$$
Sample the objective function: $y_t = f(x_t) + \epsilon_t$
Augment the data $\mathcal{D}_{1:t} = \{\mathcal{D}_{1:t-1}, (x_t, y_t)\}$ and update the GP
End

---

4. Results and Discussion

4.1. Evaluation of Machine Learning Models

In this study, a total of four machine learning models (Gaussian process regression, GPR; least squares boosting, LS-Boost; multi-layer perceptron artificial neural network, MLP-ANN; and radial basis function neural network, RBF-NN) were utilized to predict the reservoir porosity given the inputs of the acoustic travel time log (DT), bulk density log (RHOB), and Gamma-ray log (GR), using a large database of well-log records and core-porosity measurements. The Bayesian optimization algorithm was employed to find the optimum hyperparameters of (the GPR, LS-Boost, and RBF-NN) models that yield the highest prediction accuracy. Table 2 presents the hyperparameters that have been optimized along with the optimization ranges of those parameters. It should be noted that for the case of MLP-ANN, the hyperparameters were optimized within the structure of the developed in-house code, and these hyperparameters include the network weights and biases, number of hidden neurons, number of hidden layers, type of transfer function, and type of learning optimizer.

Table 3 provides the performance indicators of the developed models (GPR, LS-Boost, MLP-ANN, and RBF-NN) in terms of the root-mean-square error (RMSE), the coefficient of variation of the root-mean-square error (CVRMSE), and the coefficient of determination R^2, for the training data. It can be noted that, in general, all tested models achieved a good level of accuracy in predicting reservoir porosity, reaching coefficient-of-determination values greater than 0.9 ($R^2 > 0.90$) for the training database used. Furthermore, the GPR model gave the lowest error among all with an RMSE of 0.0093, CVRMSE of 24%, and R^2 value of 0.945. The high R-squared value and low RMSE clearly highlight the superior performance of GPR in matching the true values of reservoir porosity. The LS-Boost model was second best with an RMSE of 0.01 and R^2 of 0.935, followed by MLP-ANN and RBF-NN with RMSE of (0.0111 and 0.01165) and R^2 of 0.92 and 0.91, respectively. It should be noted that, in general, MLP-ANN and RBF-NN have similar performance for the database used.

Figure 3A–D present a cross-plot of the true reservoir porosity against the predicted reservoir porosity generated by the four machine learning models developed for this study, Gaussian process regression (GPR), least squares boosting (LS-Boost), multi-layer perceptron artificial neural network (MLP-ANN), and radial basis function neural network (RBF-NN), using training data. It can be seen from Figure 3A–D that the predicted values of reservoir porosity from these machine learning models are closely clustered around the

line of unity, which visually indicates the strong predictive capabilities of these models. Furthermore, it is evident that the outcomes of the GPR model conform closely to the unity line compared to the rest of the models, which indicates the superior performance of the GPR model used.

Table 2. Optimized values of hyperparameters for all AI models used in this study.

| Model Type	Optimizable Parameters	Optimized Value	Optimization Range
Gaussian process regression	Length scale	0.544	0.01–100
	Signal standard deviation	0.038	0.001–100
	Noise standard deviation	0.018	0.0001–1
	Kernel function type	SE	SE, Exp, M 3/2, M 5/2, RSE
Least square boosting ensemble	Minimum leaf size	1	1–300
	Learning rate	0.304	0.01–1
	Number of learning cycles	100	1–500
RBF neural network	Maximum number of neurons	80	1–100
	Spread factor	0.95	0.01–10
MLP neural network	Number of hidden layers	1	1–3
	Number of hidden neurons	15	1–30
	Type of transfer function	Log sigmoid	Tan sigmoid, Log sigmoid
	Type of learning optimizer	Levenberg–Marquardt	LM, SC, OSS

SE, square exponential; Exp, exponential; M 3/2, Matern 3/2; M 5/2, Matern 5/2; RSE, rational square exponential; RBF, LM, Levenberg–Marquardt; SC, scaled conjugate gradient; OSS, one-step secant.

Table 3. Performance indicators of the developed machine learning models for training dataset.

AI Model	R^2	RMSE	CVRMSE
			%
Gaussian process regression (GPR)	0.945	0.00932	24.61
Least square boosting ensemble (LS-Boost)	0.935	0.01014	26.78
MLP artificial neural network (MLP-ANN)	0.921	0.01113	29.39
RBF neural network (RBF-NN)	0.913	0.01165	30.75

The assessment of the model's ability to generalize was carried out by testing them on an unseen blind-testing database comprising (386) well-log records that were not included in the training/cross-validation stage. This testing process is aimed at ensuring that our developed models can perform accurately when faced with new data to which they have not been previously exposed. Table 4 provides the statistical performance indicators of each model (GPR, LS-Boost, MLP-ANN, and RBF-NN) in terms of RMSE, CVRMSE, and R^2 values using the (unseen) testing dataset. The outcomes shown in Table 4 present a performance pattern that is consistent with the previously observed training data trend. Specifically, it is apparent that the GPR model continued to demonstrate a superior performance by exhibiting a low root-mean-square error (RMSE) of 0.0105 and an impressive coefficient-of-determination value (R^2) of 0.92 as compared to the other models evaluated herein. The LS-Boost model stands second in rank with its RMSE value slightly higher than GPR at 0.01154 and an R^2 value equaling 0.91. The MLP model had a low RMSE of 0.01194, indicating reasonable predictive-accuracy capabilities overall, but lower than those exhibited by the top performer mentioned above. The RBF model was last in the line, with an R^2 of 0.9 and RMSE of 0.01213; however, these are considered acceptable, given the heterogeneous and complex nature of the carbonate reservoir under consideration. Figure 4A–D present a cross-plot of the true reservoir porosity against the predicted reservoir porosity generated by the four machine learning models developed in this study using the unseen testing dataset. The cross-plots presented in this figure confirm the findings presented earlier in Table 4, that is the GPR model exhibited the greatest accuracy, since its generated data tightly clusters around the line of unity in comparison to the other studied

models. However, Figure 4 also shows that the data produced by the other analyzed models perform well, with the bulk of the projected values clustering fairly close around the unity line, indicating little divergence from the observed values. These findings demonstrated the ability of the developed machine learning models to predict reservoir porosity accurately, as well as their ability to generalize well with new data.

(A)

(B)

(C)

(D)
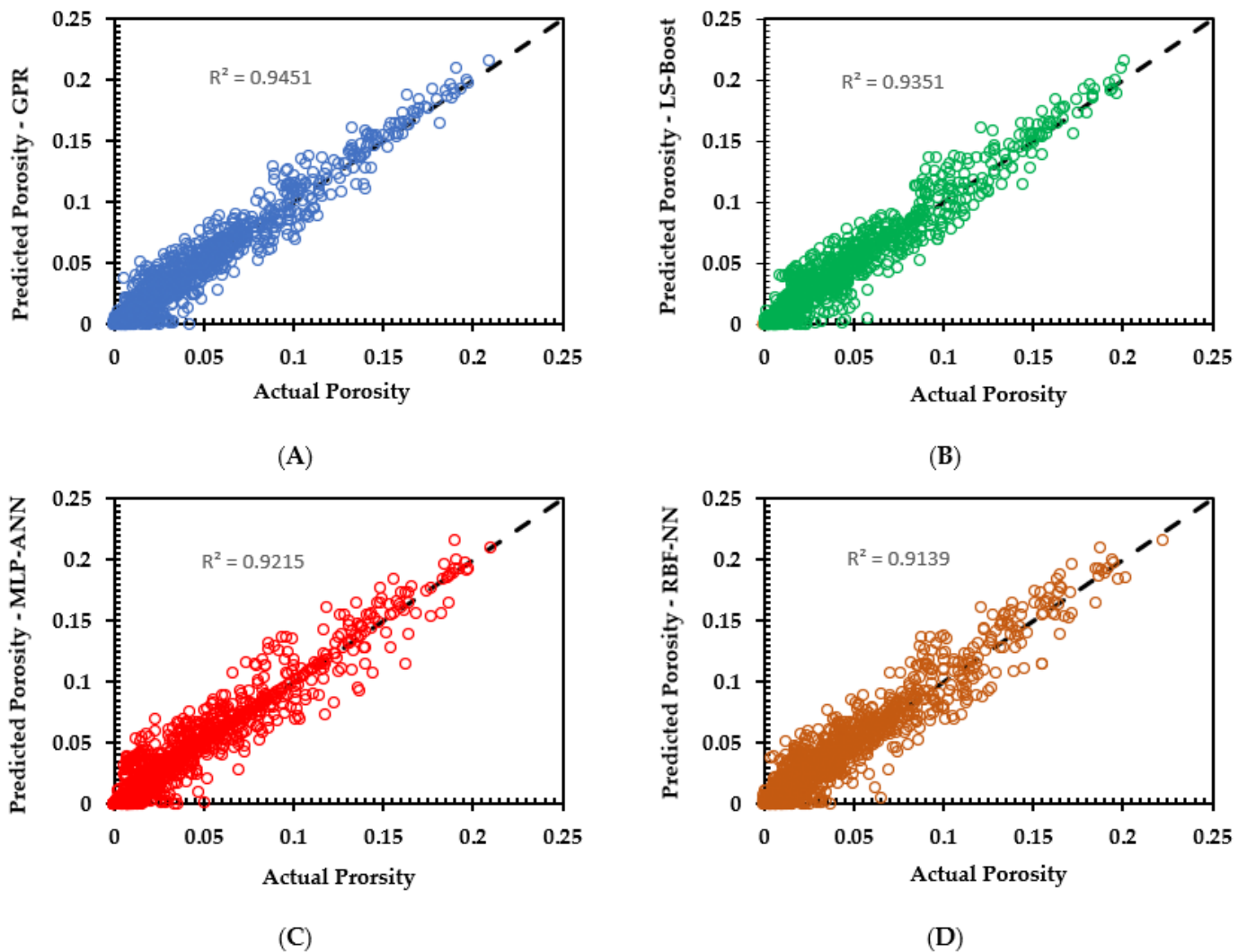
Figure 3. Cross-plot of true porosity vs. predicted porosity (using training data): (**A**) GPR model; (**B**) LS-Boost model; (**C**) MLP-ANN model; (**D**) RBF-NN model.

Table 4. Performance indicators of the developed machine learning models for (unseen) testing data.

| AI Model | R^2 | RMSE | CVRMSE |
			%
Gaussian process regression (GPR)	0.9230	0.01050	28.41
Least square boosting ensemble (LS-Boost)	0.9070	0.01154	31.24
MLP artificial neural network (MLP-ANN)	0.9004	0.01194	32.30
RBF neural network (RBF-NN)	0. 9011	0.01213	32.83
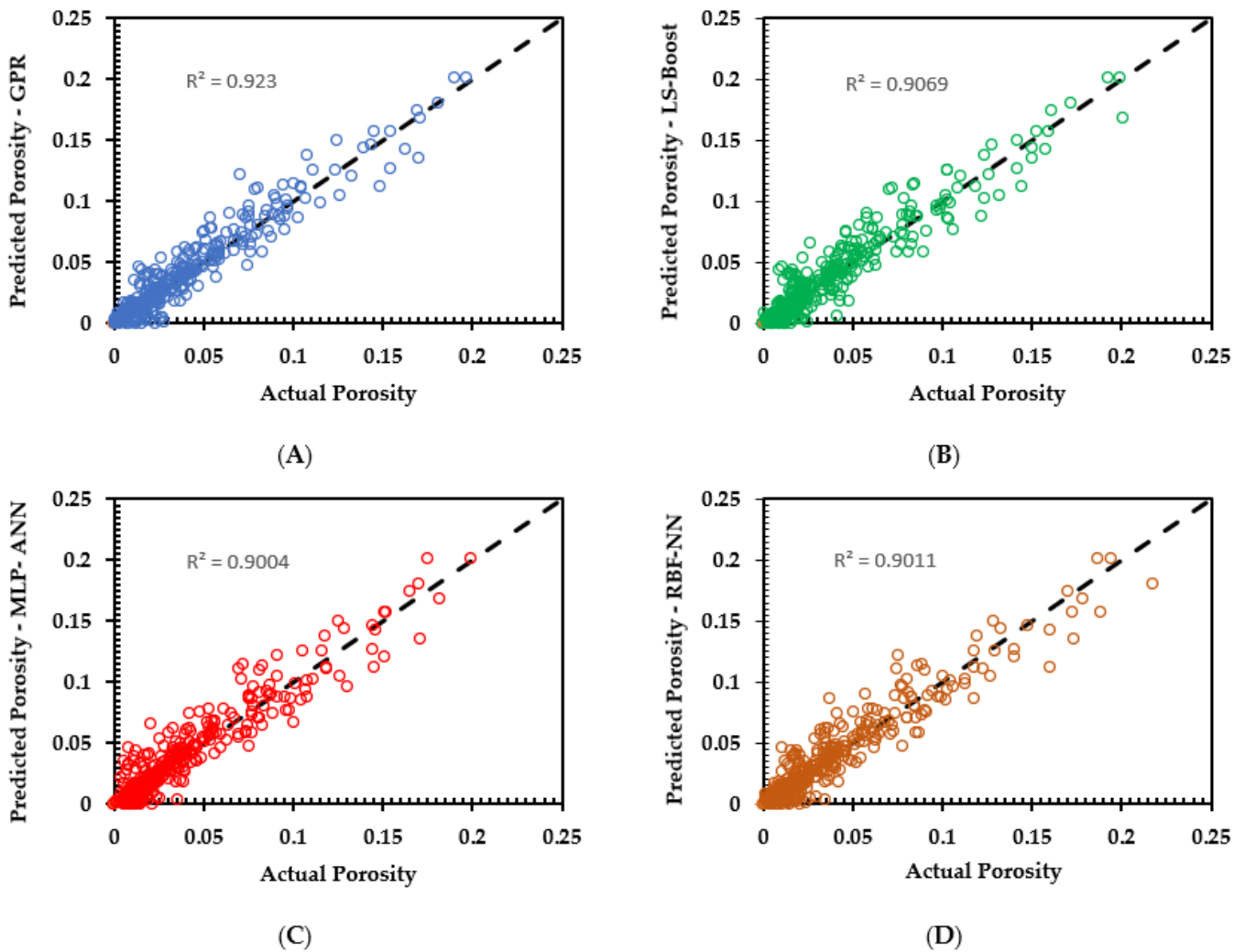
Figure 4. Cross-plot of true porosity vs. predicted porosity (using unseen testing data): (**A**) GPR model; (**B**) LS-Boost model; (**C**) MLP-ANN model; (**D**) RBF-NN model.

4.2. Development of an Explicit ANN-Based Porosity Formula

Following the encouraging results of the machine learning models presented in the previous section, we aimed at developing an AI-based explicit formula for the prediction of reservoir porosity. This was done with the help of the results of the in-house MLP-ANN model.

A trained neural network model can be converted into a mathematical equation using its weights, biases, and transfer function [52]. This formula is typically expressed as in Equation (16), which is a general formula that relates the input and output parameters for a single-hidden-layer neural network.

$$Y = f_O \left\{ b_o + \sum_{k=1}^{h} \left[w_k \times f_H \left\{ b_{hk} + \sum_{i=1}^{m} w_{ik} X_i \right\} \right] \right\} \tag{16}$$

Here, m is the number of input variables, h is the number of hidden neurons, (b_o) is the bias of the output layer; (w_k) is the weights connecting the hidden layer to the output layer, (b_{hk}) is the biases of the hidden layer, (w_{ik}) is the weights connecting the input layer to the hidden layer, (X_i) is the normalized input variable i, and (Y) is the normalized output variable; (f_H) is the transfer function of the hidden layer (i.e., sigmoid, hyperbolic tangent, or ReLU), while (f_O) is the output layer transfer function.

In this study, we have three normalized input variables (i.e., m = 3; DT_n, GR_n, and $RHOB_n$), one normalized output ($Y = \phi_n$), the number of hidden neurons is fifteen (i.e.,

h = 15), and the utilized transfer function is log-sigmoid for both the hidden layer and output layer (i.e., f_O and f_H = log-sigmoid). Accordingly, the general formula presented in Equation (16) can be rewritten as follows:

$$\phi_n = f_O\{A\} = \frac{1}{1 + \exp(-A)} \tag{17}$$

$$A = b_o + \sum_{i=1}^{15} B_i \tag{18}$$

where B values are presented in Table 5.

Table 5. Values of parameter B in Equation (18).

B_1 to B_{15} Values for Equation (18)
$B_1 = -0.643755/\{1 + \exp\left[-(17.0299DT_n - 15.688GR_n - 1.0192RHOB_n - 4.6410)\right]\}$
$B_2 = 4.319593/\{1 + \exp\left[-(-2.1639DT_n - 2.4844GR_n + 12.585RHOB_n - 2.171)\right]\}$
$B_3 = -6.346382/\{1 + \exp\left[-(-7.6639DT_n + 3.8462GR_n + 5.089RHOB_n - 6.49685)\right]\}$
$B_4 = -11.19457/\{1 + \exp\left[-(-0.7176DT_n + 3.0776GR_n + 11.8052RHOB_n - 2.4289)\right]\}$
$B_5 = -1.947745/\{1 + \exp\left[-(-8.2322DT_n + 5.77043GR_n - 10.755RHOB_n + 9.18118)\right]\}$
$B_6 = 10.670187/\{1 + \exp\left[-(3.09392DT_n - 13.828GR_n + 10.118RHOB_n - 8.42077)\right]\}$
$B_7 = 4.423562/\{1 + \exp\left[-(10.4638DT_n + 10.0488GR_n - 8.2806RHOB_n + 2.03904)\right]\}$
$B_8 = -9.137737/\{1 + \exp\left[-(-8.2019DT_n - 5.0329GR_n - 8.3817RHOB_n + 2.24147)\right]\}$
$B_9 = -6.222161/\{1 + \exp\left[-(-3.6924DT_n - 4.6341GR_n - 17.526RHOB_n + 6.48157)\right]\}$
$B_{10} = 0.094367/\{1 + \exp\left[-(-2.0643DT_n - 1.4978GR_n - 0.2585RHOB_n - 1.93168)\right]\}$
$B_{11} = 2.538843/\{1 + \exp\left[-(-3.7017DT_n - 4.9897GR_n + 0.40905RHOB_n + 1.0266)\right]\}$
$B_{12} = 4.865178/\{1 + \exp\left[-(4.0603DT_n + 3.142GR_n - 10.46215RHOB_n + 1.86999)\right]\}$
$B_{13} = -1.599128/\{1 + \exp\left[-(-4.7663DT_n + 11.7106GR_n - 25.15RHOB_n + 11.05048)\right]\}$
$B_{14} = -0.245659/\{1 + \exp\left[-(-2.101DT_n - 1.96202GR_n + 0.46881RHOB_n - 1.5497)\right]\}$
$B_{15} = -13.071971/\{1 + \exp\left[-(3.05871DT_n - 16.31GR_n + 5.42428RHOB_n - 5.16961)\right]\}$
$b_o = 2.594365$

DT_n, normalized acoustic travel time log; GR_n, normalized Gamma ray log; $RHOB_n$, normalized bulk density log.

A brief demonstration of the use of the proposed ANN-based porosity correlation is presented in Appendix A.

It should be noted that Equation (18) simply presents the MLP-network as a linear sum of sigmoidal functions (B_1 to B_{15} in Table 5), which can be conveniently calculated using a simple spreadsheet or any other suitable mathematical platform. Moreover, it is worth noting that the coefficients of functions (B_1 to B_{15}) are nothing but the optimized weights and biases of the in-house MLP-ANN model presented in Table 6.

Table 6. Optimized values of weights and biases for the developed MLP-ANN model.

Hidden Neuron	Weights (W_1)			Hidden Layer Bias (b_h)	Weights (W_2)	Output Layer Bias (b_o)
	DT_n	GR_n	$ROHB_n$			
1	17.029976	−15.687875	−1.019189	−4.640982	−0.643755	2.594365
2	−2.163948	−2.484408	12.584483	−2.171149	4.319593	
3	−7.663929	3.846159	5.089029	−6.496849	−6.346382	
4	−0.717566	3.0776	11.805209	−2.428945	−11.194565	
5	−8.232247	5.770426	−10.77542	9.181175	−1.947745	
6	3.093922	−13.828028	10.111827	−8.420765	10.670187	
7	10.463783	10.048819	−8.280626	2.039037	4.423562	
8	−8.201936	−5.032945	−8.381655	2.241477	−9.137737	
9	−3.692441	−4.634098	−17.525692	6.481565	−6.222161	
10	−2.064329	−1.497829	−0.258506	−1.93168	0.094367	
11	−3.70173	−4.989678	0.409045	1.026638	2.538843	

Table 6. *Cont.*

Hidden Neuron	Weights (W_1)			Hidden Layer Bias (b_h)	Weights (W_2)	Output Layer Bias (b_o)
	DT_n	GR_n	$ROHB_n$			
12	4.060295	3.141995	−10.462156	1.869994	4.865178	
13	−4.766314	11.710604	−25.149687	11.050483	−1.599128	
14	−2.100971	−1.962021	0.468808	−1.549698	−0.245659	
15	3.058714	−16.310364	5.42428	−5.169609	−13.071971	

W_1, weights connecting input layer to hidden layer; W_2, weights connecting hidden layer to output layer.

Furthermore, it should be noted that Equation (17) provides the normalized porosity value (ϕ_n); therefore, in order to obtain the true porosity value (ϕ), Equation (19) must be used to de-normalize the result as follows:

$$\phi = \phi_n \times (\phi_{max} - \phi_{min}) + \phi_{min} \tag{19}$$

where ϕ_{max} and ϕ_{min} are the maximum and minimum porosity values used in this study, referring to Table 1.

The testing dataset (the 386 well-log records used in Section 4.1 to evaluate the model's generalization ability) were utilized for comparing Equation (17) and the MPL-ANN model outputs. Figure 5 displays a cross-plot of the normalized porosity values from Equation (17) against the normalized porosity from the in-house MPL-ANN code using the testing dataset. It is clear from Figure 5 that all of the points perfectly fit the line of unity, indicating that the developed equation (Equation (17)) provides values that are nearly identical to those from the in-house MLP-ANN code.
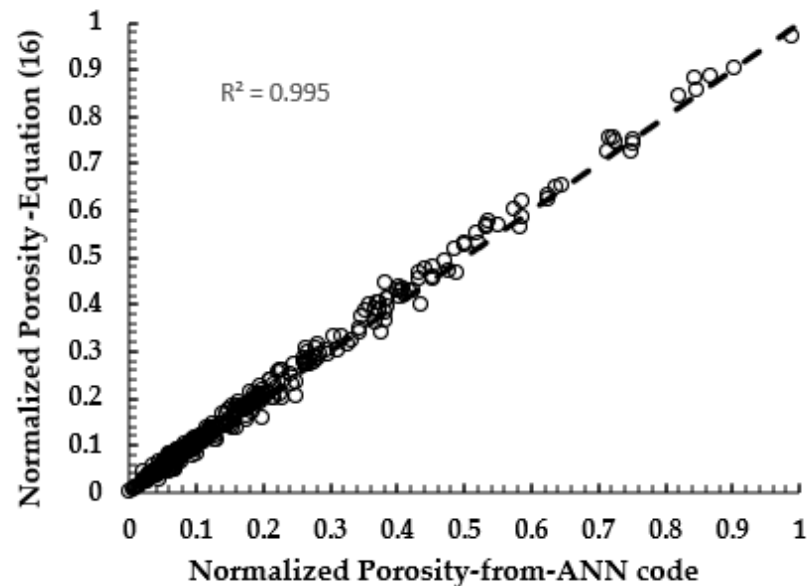

Figure 5. Cross-plot of the normalized porosity values from Equation (17) against the normalized porosity from the MLP-ANN code.

Furthermore, an independent dataset of 205 well-log records from well X, which was not used in any phase of this study (i.e., training, cross-validation, or testing), was utilized to further assess the accuracy of the developed equation (Equation (17)). Figure 6A presents the reservoir's actual porosity profile of well X against the porosity profile predicted by Equation (17) for the independent dataset. It can be seen from Figure 6A that the developed equation (Equation (17)) produces a porosity profile that closely matches the true porosity profile. In addition, Figure 6B shows a cross-plot of estimated porosity (from Equation (17)) versus actual values, and it can be seen that most of the data cluster tightly around the

unity line, indicating minimal divergence between predicted and true values with a low RMSE of (0.01) and a high coefficient-of-determination R^2 of 0.9. Such encouraging results demonstrate the potential of using the ANN-based mathematical model (Equation (17)) as an accurate and easy-to-use tool for predicting reservoir porosity without any prior experience with utilizing or implementing machine learning models, such as MLP-ANN. However, it is important to note that the developed model (Equation (17)) should be used within the studied range of pertinent parameters and that caution should be exercised when extrapolating beyond this range. Having said that, it is always possible to improve and calibrate the developed model by incorporating more new data with various geological settings. The new data can be used to retrain the in-house code and recalculate the optimum network weights and biases, which then can be used to update the developed equation (Equation (17)).
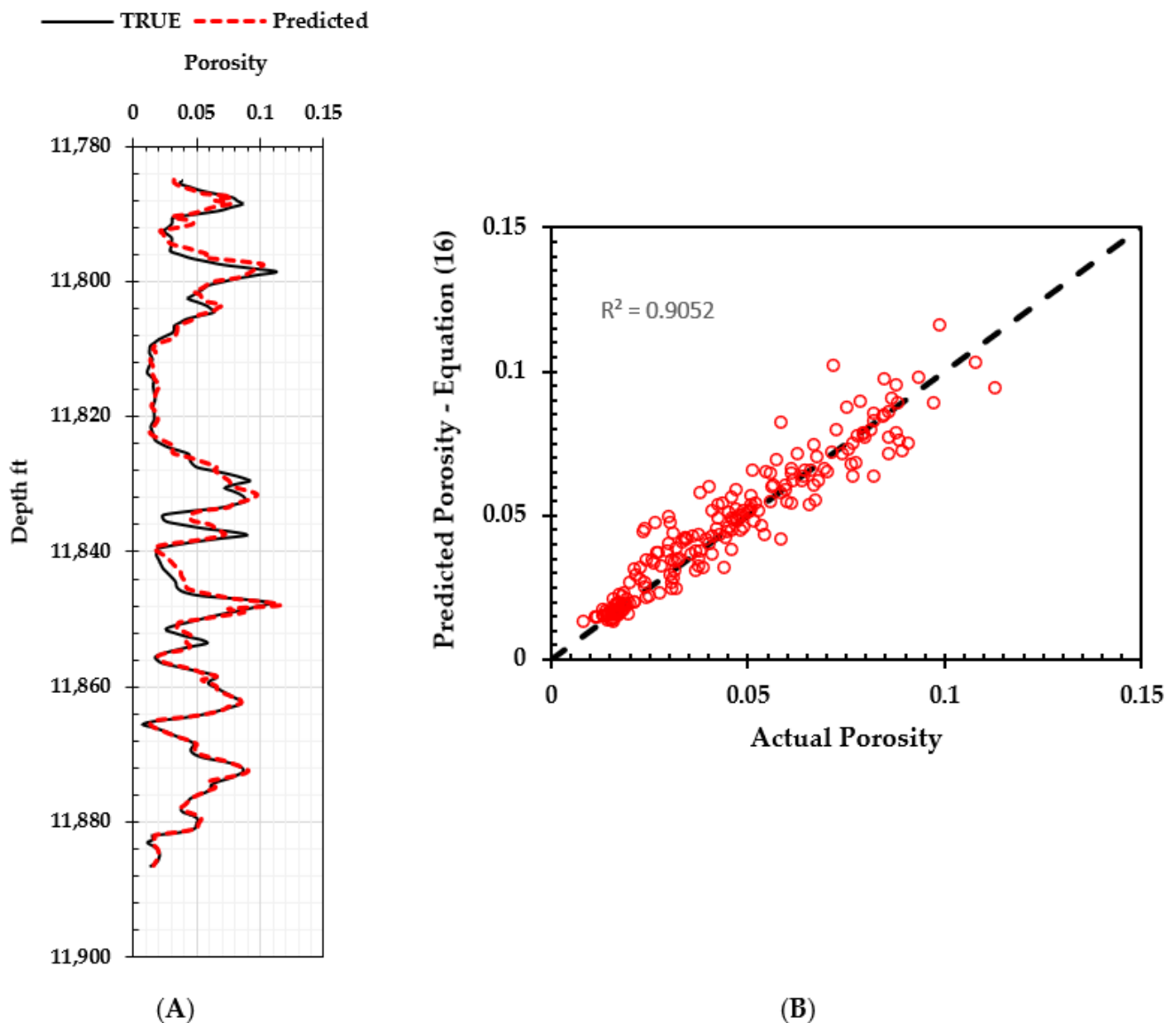
Figure 6. (**A**) The true porosity profile of well X presented against the porosity profile predicted by Equation (17) for the independent dataset; (**B**) cross-plot of estimated porosity (Equation (17)) versus actual porosity values for the independent dataset.

4.3. Comparison of the Developed Mathematical Model with Existing Correlations

The developed ANN-based mathematical model was compared with two empirical correlations used in the oil industry for estimating porosity from acoustic-travel-time sonic

logs. These correlations are the Wyllie correlation [1] and the Raymer correlation [2]. Unlike the developed ANN-based mathematical model, both correlations require prior knowledge of the lithology type (e.g., sandstone, carbonate, or any other reservoir rock type) and the pore fluid type (e.g., fresh water or salted water). The mathematical form of both correlations is as follows:
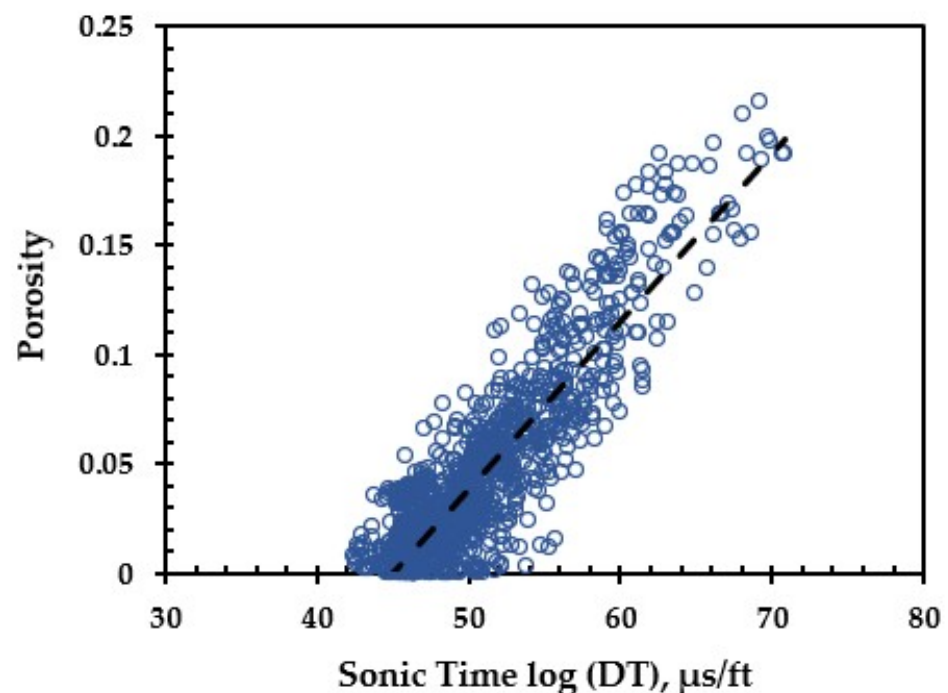
Wyllie correlation,

$$\phi = \frac{\Delta t - \Delta t_{matrix}}{\Delta t_{fluid} - \Delta t_{matrix}} \tag{20}$$

Raymer correlation,

$$\phi = 0.625 \times \left[1 - \frac{\Delta t_{matrix}}{\Delta t} \right] \tag{21}$$

where Δt_{matrix} is the transient time for the matrix (μs/ft) and Δt_{fluid} is the transient time for the pore fluid (μs/ft).

Drawing a cross-plot of reservoir porosity vs. transit time (Δt) at the same depth is a typical practice in the industry. This should result in a straight line that can be extrapolated to the *x*-axis to provide a value for the local matrix transit time (Δt_{matrix}). As a result, we applied this to our data, and the extrapolation produced the matrix transient time (Δt_{matrix} = 45 μs/ft), as shown in Figure 7. This (Δt_{matrix}) value is in the range of the normal values for limestone (47.6 s/ft) and dolomite (43.5 s/ft), which is consistent with the geological description of the carbonate reservoir under consideration.


Figure 7. Cross-plot of reservoir porosity vs. transit sonic time (Δt) at the same depth, the dotted black line is the best-fit line of the porosity data.

Figure 8A–D present cross-plots of the porosity predicted by the Wyllie correlation and the Raymer correlation against the reservoir's actual porosity for both the training and testing datasets. It can be noted that, in comparison to what we have previously shown with the machine learning models, neither correlation generally obtained high accuracy for the dataset employed. However, the Wyllie correlation performed better than the Raymer correlation, with an R^2 value of 0.81 for the training data compared to an R^2 of 0.78 for the Raymer correlation, and an R^2 value of 0.84 for the testing data compared to 0.82 for the Raymer correlation. The statistical performance indicators of the Wyllie correlation, the Raymer correlation, the ANN-based correlation, and the GPR model are

shown in Table 7 for the training and testing datasets. It can be noted that the GPR model demonstrated a superior performance by exhibiting low RMSE values (0.009 for training data and 0.0105 for testing data) and remarkable R-squared values (0.94 for training data and 0.92 for testing data), as compared to the ANN-based, Wyllie, and Raymer correlations. Furthermore, it can be seen from the results in Table 7 that the ANN-based correlation outperformed both the Wyllie and Raymer correlations with an RMSE of 0.011 and an R^2 of 0.92 for the training data, compared to an RMSE of 0.018 and an R^2 of 0.82 for the Wyllie correlation, and an RMSE of 0.029 and an R^2 of 0.78 for the Raymer correlation. A high R^2 value of 0.9 was also achieved for the testing data by the ANN-based correlation, as opposed to R^2 values of 0.84 and 0.82 for the Wyllie correlation and the Raymer correlation, respectively. Such findings show the superiority of the ANN-based mathematical model compared to commonly used porosity correlations.
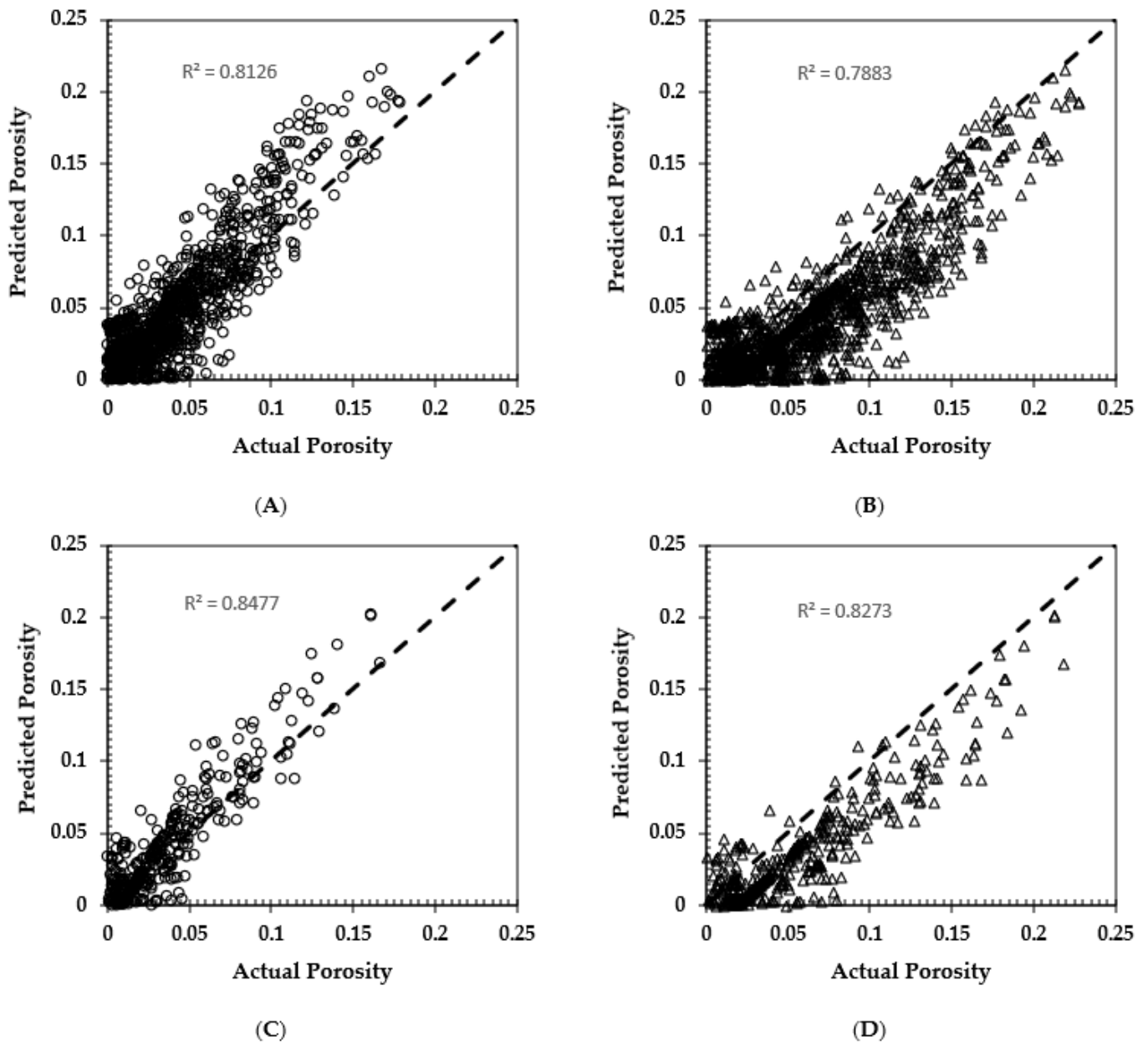
Figure 8. Cross-plot of the porosity predicted by Wyllie correlation and Raymer correlation against reservoir true porosity: (**A**) Wyllie correlation—training data; (**B**) Raymer correlation—training data; (**C**) Wyllie correlation—testing data; (**D**) Raymer correlation—testing data.

Table 7. Performance indicators of Wyllie and Raymer correlations and Equation (17) for training and testing data.

AI Model	Training Data			Testing Data		
	R^2	RMSE	CVRSME %	R^2	RMSE	CVRSME %
Gaussian process regression (GPR)	0.94	0.009	24	0.92	0.0105	28
MLP-ANN-based model (Equation (17))	0.92	0.011	29	0.90	0.0119	32
Wyllie correlation	0.82	0.018	47	0.84	0.0155	41
Raymer correlation	0.78	0.029	77	0.82	0.028	75

Overall, the results of this study demonstrated the potential of machine learning models in accurately predicting reservoir porosity, which can greatly benefit the oil and gas industry in terms of more efficient exploration and production of hydrocarbon reserves. These findings suggest that machine learning techniques can be highly effective in optimizing production and reducing the exploration costs in the oil and gas industry. Further research in this area could focus on incorporating additional data sources or refining the input parameters to improve the predictive accuracy of these models even further, potentially leading to a more efficient and cost-effective exploration and field-development processes.

5. Conclusions

Four cutting-edge artificial intelligence (AI) models, including the multilayer perceptron artificial neural network (MLP ANN), Gaussian process regression (GPR), least squares gradient boosting ensemble (LS-Boost), and radial basis function neural network (RBF-NN), were combined with more than 2000 conventional well-log records to accurately predict reservoir porosity. All the artificial intelligence models showed an impressive degree of accuracy, reaching R-squared values greater than 90% for both the training and blind-testing stages. When compared to the other studied AI models, the GPR model performed better in terms of R-squared values, root-mean-square error (RMSE), and coefficient of variation of the root-mean-squared error (CVRMSE).

Furthermore, an easy-to-use approach for forecasting porosity based on conventional well-log data was developed using an AI-based explicit correlation derived from an in-house MLP artificial neural network (ANN) model. R-squared scores of 92% and 90% for the corresponding training and blind-testing datasets used in this study showed that the ANN-based correlation exhibits a surprisingly promising level of accuracy. Moreover, for the input data used in this investigation, the Wyllie and Raymer correlations, which are often used in the industry, showed insufficient porosity predictions compared to the developed ANN-based correlation.

In summary, the findings of this study demonstrated the high potential of machine learning methods for accurately predicting the porosity of carbonate reservoirs. However, it is important to note that the accuracy of the porosity estimation using these machine learning techniques heavily relies on the quality and quantity of the well-log data used as inputs. Therefore, it is crucial to ensure that data-collection processes be carefully designed and executed and that the data used for training the machine learning models are representative of the subsurface conditions. Furthermore, ongoing monitoring and calibration of the developed machine learning models with new data are necessary to ensure that they continue to provide reliable and accurate predictions over time.

Author Contributions: Conceptualization, S.A. and R.A.A.; methodology, S.A. and R.A.A.; software, S.A. and R.A.A.; validation, R.A.A. and A.A.; formal analysis, S.A.; investigation, S.A.; resources, A.A. and G.H.; data curation, G.H.; writing—original draft preparation, S.A.; writing—review and editing, R.A.A., S.A. and A.A.; visualization, S.A.; supervision, G.H.; All authors have read and agreed to the published version of the manuscript.

Appendix A

In this section, we present a brief demonstration of the use of the proposed ANN-based porosity correlation (Equation (17) in the main text).

$$\phi_n = \frac{1}{1 + \exp(-A)} \tag{A1}$$

$$A = b_o + \sum_{i=1}^{15} B_i \tag{A2}$$

where the *B* values are presented in Table 5 in Section 4.2.

Table A1 presents some input data from the porosity database used in this study. To perform the calculation, take the first data point set in Table A1. The calculation steps can be summarized as follows:

1. Use the normalized input data in the first row in Table A1: $DT_n = 0.771185479$, $GR_n = 0.121122333$, and $RHOB_n = 0.395517115$.
2. Direct application of these input data in Table 5 (in Section 4.2 of the main text) will give the following values for the sigmoidal functions B_1 to B_{15}: $B_1 = -0.642436837$, $B_2 = 3.013905339$, $B_3 = -0.000309464$, $B_4 = -9.928624721$, $B_5 = -0.633246848$, $B_6 = 0.254874861$, $B_7 = 4.422151675$, $B_8 = -0.003038635$, $B_9 = -0.128510023$, $B_{10} = 0.002050128$, $B_{11} = 0.23759064$, $B_{12} = 3.776504233$, $B_{13} = -0.383421685$, $B_{14} = -0.009418638$, and $B_{15} = -0.869945291$. It should be noted that (*bo*) in Equation (A2) is the bias of the output layer, which is always constant regardless of input data (*bo* = 2.594365).
3. Calculate (*A*) from Equation (A2) using the results of B_1 to B_{15} and *bo* in Step 2; application of Equation (A2) will give a value of (*A* = 1.702489735).
4. Calculate the normalized porosity (ϕ_n) from Equation (A1) using the (*A*) value from Step 3; application of Equation (A1) will give a value of $\phi_n = 0.845859629$.
5. Finally, the actual porosity can be obtained by denormalizing (ϕ_n) using Equation (19) in the main text.

Table A1. A random sample from the database used in this study.

Sample ID	Normalized Input Data			Normalized Output (ϕ_n)	(ϕ_n) from Equation (17)
	DT_n	GR_n	$RHOB_n$		
1	0.771185479	0.121122333	0.395517115	0.86162748	0.845859629
2	0.257402155	0.274615363	0.751833354	0.20437786	0.201431142
3	0.235696729	0.413042873	0.817785358	0.153756067	0.150223503
4	0.249877103	0.260050636	0.911901571	0.185887829	0.180857764
5	0.529060314	0.094524664	0.492898039	0.482527269	0.466471079
6	0.362261297	0.229766575	0.364526739	0.289830979	0.298473779
7	0.858725657	0.26012019	0.454184432	0.894637593	0.883088142
8	0.259255058	0.269718721	0.761554392	0.20733547	0.202935626
9	0.337530724	0.372478647	0.804507248	0.31965521	0.329507671
10	0.74316506	0.172258854	0.014715556	0.726599641	0.72330119

These input data are being made available in Table A1 for readers to be able to perform a hands-on application by following the same procedure shown in the previous steps.

References

1. Wyllie, M.R.J.; Gregory, A.R.; Gardner, L.W. Elastic Wave Velocities in Heterogeneous and Porous Media. *Geophysics* **1956**, *21*, 41–70. [CrossRef]
2. Raymer, L.L.; Hunt, E.R.; Gardner, J.S. An Improved Sonic Transit Time-to-Porosity Transform. In Proceedings of the SPWLA 21st Annual Logging Symposium 1980, SPWLA-1980-P, Lafayette, LA, USA, 8–11 July 1980.
3. Wendt, W.A.; Sakurai, S.; Nelson, P.H. Permeability Prediction from Well Logs Using Multiple Regression. In *Reservoir Characterization*; Elsevier: Amsterdam, The Netherlands, 1986; pp. 181–221.
4. Jensen, J.L.; Lake, L.W. Optimization of Regression-Based Porosity-Permeability Predictions. In *Transactions of the 10th Formation Evaluation Symposium*; CWLS: Calgary, AB, Canada, 1985.
5. Amaefule, J.; Altunbay, M.; Tiab, D.; Kersey, D.; Keelan, D. Enhanced Reservoir Description: Using Core and Log Data to Identify Hydraulic (Flow) Units and Predict Permeability in Uncored Intervals/Wells. In *Proceedings of the SPE Annual Technical Conference and Exhibition, Houston, TX, USA, 3–6 October 1993*; Society of Petroleum Engineers: Houston, TX, USA, 1993.
6. Xue, G.; Datta-Gupta, A.; Valko, P.; Balsingame, T. Optimal Transformations for Multiple Regression: Application to Permeability Estimation from Well Logs. In Proceedings of the SPE 35412 Improved Oil Recovery Symposium, Tulsa, OK, USA, 21–24 April 1996.
7. Mohaghegh, S.; Balan, B.; Ameri, S. Permeability Determination from Well Log Data. *SPE Form. Eval.* **1997**, *12*, 170–174. [CrossRef]
8. Datta-Gupta, A.; Xue, G.; Lee, S.H. *Nonparametric Transformations for Data Correlation and Integration: From Theory to Practice, Reservoir Characterization: RecentAdvances, AAPG Datapages*; AAPG Datapages: Tulsa, OK, USA, 1999.
9. Delfiner, P. Three Statistical Pitfalls of Phi-k Transforms. *SPE Reserv. Eval. Eng.* **2007**, *10*, 609–617. [CrossRef]
10. Bahaloo, S.; Mehrizadeh, M.; Najafi-Marghmaleki, A. Review of Application of Artificial Intelligence Techniques in Petroleum Operations. *Pet. Res.* **2022**, *in press*. [CrossRef]
11. Tariq, Z.; Aljawad, M.S.; Hasan, A.; Murtaza, M.; Mohammed, E.; El-Husseiny, A.; Alarifi, S.A.; Mahmoud, M.; Abdulraheem, A. A Systematic Review of Data Science and Machine Learning Applications to the Oil and Gas Industry. *J. Pet. Explor. Prod. Technol.* **2021**, *11*, 4339–4374. [CrossRef]
12. Wood, D.A. Predicting Porosity, Permeability and Water Saturation Applying an Optimized Nearest-Neighbour, Machine-Learning and Data-Mining Network of Well-Log Data. *J. Pet. Sci. Eng.* **2020**, *184*, 106587. [CrossRef]
13. Anifowose, F.; Abdulraheem, A. Fuzzy Logic-Driven and SVM-Driven Hybrid Computational Intelligence Models Applied to Oil and Gas Reservoir Characterization. *J. Nat. Gas Sci. Eng.* **2011**, *3*, 505–517. [CrossRef]
14. Singh, S.; Kanli, A.I.; Sevgen, S. A General Approach for Porosity Estimation Using Artificial Neural Network Method: A Case Study from Kansas Gas Field. *Stud. Geophys. Geod.* **2016**, *60*, 130–140. [CrossRef]
15. Zhang, Z.; Zhang, H.; Li, J.; Cai, Z. Permeability and Porosity Prediction Using Logging Data in a Heterogeneous Dolomite Reservoir: An Integrated Approach. *J. Nat. Gas Sci. Eng.* **2021**, *86*, 103743. [CrossRef]
16. Saggaf, M.M.; Toksöz, M.N.; Mustafa, H.M. Estimation of Reservoir Properties from Seismic Data by Smooth Neural Networks. *Geophysics* **2003**, *68*, 1969–1983. [CrossRef]
17. Yang, N.; Li, G.; Zhao, P.; Zhang, J.; Zhao, D. Porosity Prediction from Pre-Stack Seismic Data via a Data-Driven Approach. *J. Appl. Geophy.* **2023**, *211*, 104947. [CrossRef]
18. Mohaghegh, S.; Arefi, R.; Ameri, S.; Aminiand, K.; Nutter, R. Petroleum Reservoir Characterization with the Aid of Artificial Neural Networks. *J. Pet. Sci. Eng.* **1996**, *16*, 263–274. [CrossRef]
19. Helle, H.B.; Bhatt, A.; Ursin, B. Porosity and Permeability Prediction from Wireline Logs Using Artificial Neural Networks: A North Sea Case Study: Pore-Perm Prediction by Neural Nets. *Geophys. Prospect.* **2001**, *49*, 431–444. [CrossRef]
20. Hamada, G.; Elshafei, M. Neural Network Prediction of Porosity and Permeability of Heterogeneous Gas Sand Reservoirs Using NMR Logging Data. In *Proceedings of the SPE Reservoir Characterization and Simulation Conference and Exhibition, Abu Dhabi, United Arab Emirates, 16–18 September 2013*; Society of Petroleum Engineers: Houston, TX, USA, 2013.
21. Lim, J.-S.; Kim, J. Reservoir Porosity and Permeability Estimation from Well Logs Using Fuzzy Logic and Neural Networks. In *Proceedings of the SPE, Asia Pacific Oil and Gas Conference and Exhibition, Perth, Australia, 18–20 October 2004*; Society of Petroleum Engineers: Houston, TX, USA, 2004.
22. Moghadam, J.N.; Salahshoor, K. Kharrat Intelligent Prediction of Porosity and Permeability from Well Logs for an Iranian Fractured Carbonate Reservoir Petrol. *Sci. Technol.* **2011**, *29*, 2095–2112.
23. Zargari, H.; Poordad, S.; Kharrat, R. Porosity and Permeability Prediction Based on Computational Intelligences as Artificial Neural Networks (ANNs) and Adaptive Neuro-Fuzzy Inference Systems (ANFIS) in Southern Carbonate Reservoir of Iran. *Pet. Sci. Technol.* **2013**, *31*, 1066–1077. [CrossRef]
24. Al-Sabaa, A.; Gamal, H.; Elkatatny, S. Generation of a Complete Profile for Porosity Log While Drilling Complex Lithology by Employing the Artificial Intelligence. In Proceedings of the SPE Symposium: Artificial Intelligence—Towards a Resilient and Efficient Energy Industry, Virtual, 18–19 October 2021.
25. Al-Anazi, A.F.; Gates, I.D. Support Vector Regression for Porosity Prediction in a Heterogeneous Reservoir: A Comparative Study. *Comput. Geosci.* **2010**, *36*, 1494–1503. [CrossRef]
26. Zhong, Z.; Carr, T.R. Application of a New Hybrid Particle Swarm Optimization-Mixed Kernels Function-Based Support Vector Machine Model for Reservoir Porosity Prediction: A Case Study in Jacksonburg-Stringtown Oil Field, West Virginia, USA. *Interpretation* **2019**, *7*, T97–T112. [CrossRef]

27. Elkatatny, S.; Tariq, Z.; Mahmoud, M.; Abdulraheem, A. New Insights into Porosity Determination Using Artificial Intelligence Techniques for Carbonate Reservoirs. *Petroleum* **2018**, *4*, 408–418. [CrossRef]

28. Wang, B.; Wang, X.; Chen, Z. A Hybrid Framework for Reservoir Characterization Using Fuzzy Ranking and an Artificial Neural Network. *Comput. Geosci.* **2013**, *57*, 1–10. [CrossRef]

29. Zerrouki, A.A.; Aifa, T. Baddari Prediction of Natural Fracture Porosity from Well Log Data by Means of Fuzzy Ranking and an Artificial Neural Network in Hassi Messaoud Oil Field. *Alger. J. Petrol. Sci. Eng.* **2014**, *115*, 78–89. [CrossRef]

30. Anifowose, F.; Labadin, J.; Abdulraheem, A. A Least Square-Driven Functional Networks Type-2 Fuzzy Logic Hybrid Model for Efficient Petroleum Reservoir Properties Prediction. *Neural Comput. Appl.* **2013**, *23*, 179–190. [CrossRef]

31. Huang, Y.; Gedeon, T.D. Wong An Integrated Neural-Fuzzy-Genetic-Algorithm Using Hyper-Surface Membership Functions to Predict Permeability in Petroleum Reservoirs. *Eng. Appl. Artif. Intell.* **2001**, *14*, 15–21. [CrossRef]

32. Shokir, E.M.E.-M. A Novel Model for Permeability Prediction in Uncored Wells. *SPE Reserv. Eval. Eng.* **2006**, *9*, 266–273. [CrossRef]

33. Aïfa, T.; Baouche, R. Baddari Neuro-Fuzzy System to Predict Permeability and Porosity from Well Log Data: A Case Study of Hassi R' Mel Gas Field. *Alger. J. Petrol. Sci. Eng.* **2014**, *123*, 217–229. [CrossRef]

34. Abdel Azim, R. Prediction of Multiphase Flow Rate for Artificially Flowing Wells Using Rigorous Artificial Neural Network Technique. *Flow Meas. Instrum.* **2020**, *76*, 101835. [CrossRef]

35. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; The MIT Press: New York, NY, USA, 2006.

36. Nguyen-Tuong, D.; Seeger, M.; Peters, J. Model Learning with Local Gaussian Process Regression. *Adv. Robot.* **2009**, *23*, 2015–2034. [CrossRef]

37. Alajmi, M.S.; Almeshal, A.M. Modeling of Cutting Force in the Turning of AISI 4340 Using Gaussian Process Regression Algorithm. *Appl. Sci.* **2021**, *11*, 4055. [CrossRef]

38. Alatefi, S.; Almeshal, A.M. A New Model for Estimation of Bubble Point Pressure Using a Bayesian Optimized Least Square Gradient Boosting Ensemble. *Energies* **2021**, *14*, 2653. [CrossRef]

39. Ribeiro, M.H.D.M.; Dos Santos Coelho, L. Ensemble Approach Based on Bagging, Boosting and Stacking for Short-Term Pre-Diction in Agribusiness Time Series. *Appl. Soft Comput.* **2020**, *86*, 105837. [CrossRef]

40. Anifowose, F.; Labadin, J.; Abdulraheem, A. Prediction of Petroleum Reservoir Characterization with a Stacked Generalization Ensemble Model of Support Vector Machines. *Appl. Soft Comput.* **2014**, *26*, 483–496. [CrossRef]

41. Helmy, T.; Rahman, S.M.; Hossain, M.I.; Abdelraheem, A. Non-Linear Heterogeneous Ensemble Model for Permeability Prediction of Oil Reservoirs. *Arab. J. Sci. Eng.* **2013**, *38*, 1379–1395. [CrossRef]

42. Qureshi, A.S.; Khan, A.; Zameer, A.; Usman, A. Wind Power Prediction Using Deep Neural Network Based Meta Regression and Transfer Learning. *Appl. Soft Comput.* **2017**, *58*, 742–755. [CrossRef]

43. Friedman, J.H. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]

44. Haykin, S. *Neural Networks and Learning Machines*; Prentice Hall: Hoboken, NJ, USA, 2009.

45. Dutta, P.; Pratihar, D.K. Modeling of TIG Welding Process Using Conventional Regression Analysis and Neural Network-Based Approaches. *J. Mater. Process. Technol.* **2007**, *184*, 56–68. [CrossRef]

46. Hashemi Fath, A.; Madanifar, F.; Abbasi, M. Implementation of Multilayer Perceptron (MLP) and Radial Basis Function (RBF) Neural Networks to Predict Solution Gas-Oil Ratio of Crude Oil Systems. *Petroleum* **2020**, *6*, 80–91. [CrossRef]

47. Abdel Azim, R.; Aljehani, A. Neural Network Model for Permeability Prediction from Reservoir Well Logs. *Processes* **2022**, *10*, 2587. [CrossRef]

48. Abdel-Azim, R. Estimation of Bubble Point Pressure and Solution Gas Oil Ratio Using Artificial Neural Network. *Int. J. Ther.* **2022**, *14*, 100159. [CrossRef]

49. Abdel Azim, R. Application of Artificial Neural Network in Optimizing the Drilling Rate of Penetration of Western Desert Egyptian Wells. *SN Appl. Sci.* **2020**, *2*, 1177. [CrossRef]

50. Chen, S.; Cowan, C.N.; Grant, P.M. Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks. *IEEE Trans. Neural Netw.* **1991**, *2*, 302–309. [CrossRef]

51. Brochu, E.; Cora, V.M.; De Freitas, N. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. *arXiv* **2010**, arXiv:1012.2599.

52. Lawal, A.I.; Idris, M.A. An Artificial Neural Network-Based Mathematical Model for the Prediction of Blast-Induced Ground Vibrations. *Int. J. Environ. Stud.* **2020**, *77*, 318–334. [CrossRef]